

# HT82A525R DMA Interface 在指紋辨識的運用

文件編碼：HA0251T

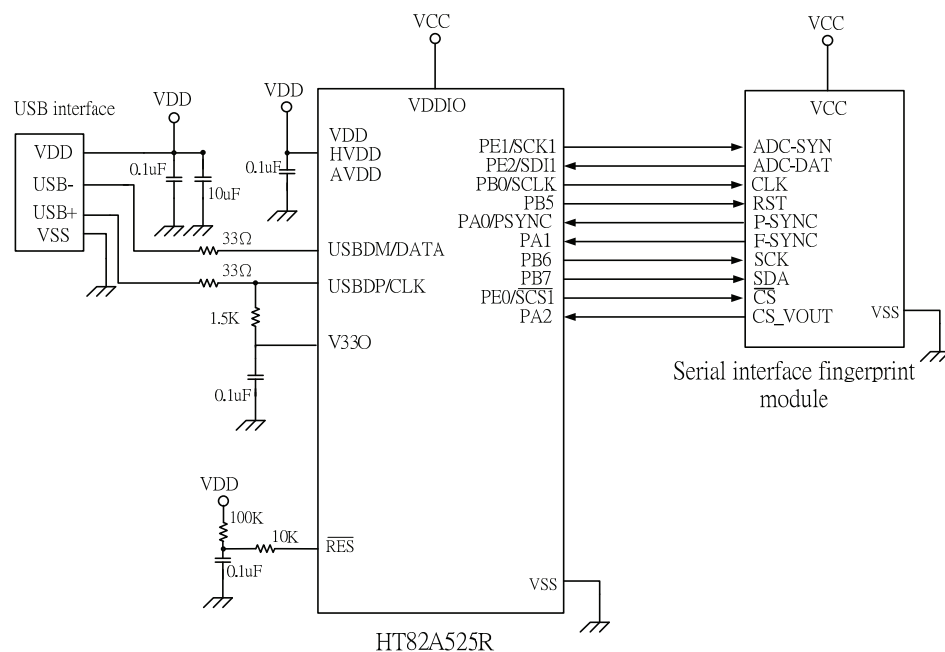
## 簡介

本文介紹如何運用 HT82A525R 控制指紋辨識模組及如何使用 HT82A525R DMA 介面快速的接收指紋辨識模組產生的大量資料，HT82A525R 包含了 SPI 串列 DMA 介面和 8 bits 並列 DMA 介面。本文將分別舉例串列 DMA 介面指紋辨識模組和並列 DMA 介面指紋辨識模組的運用，著重考慮如何控制 DMA 介面並且利用 USB FIFO 傳送至 USB 介面。

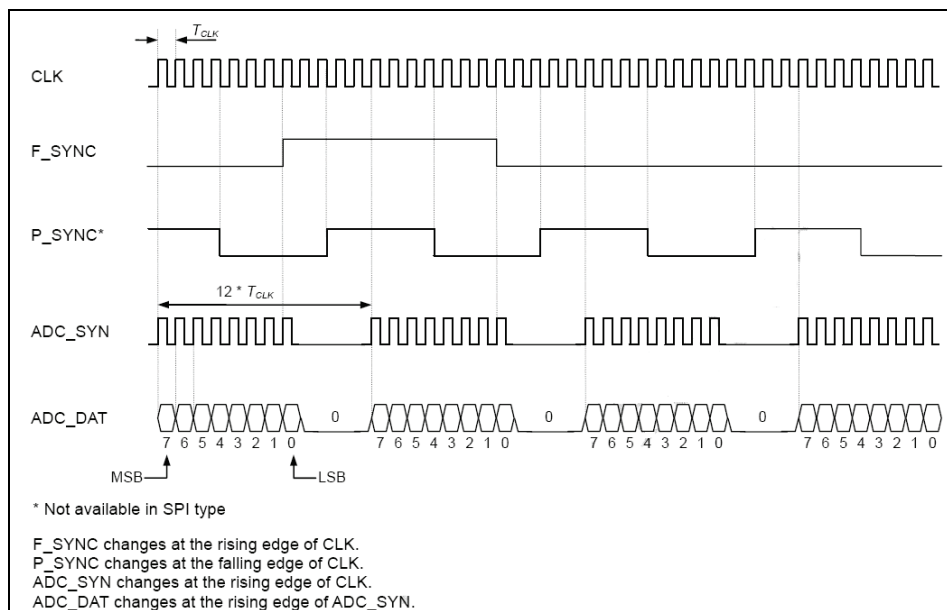
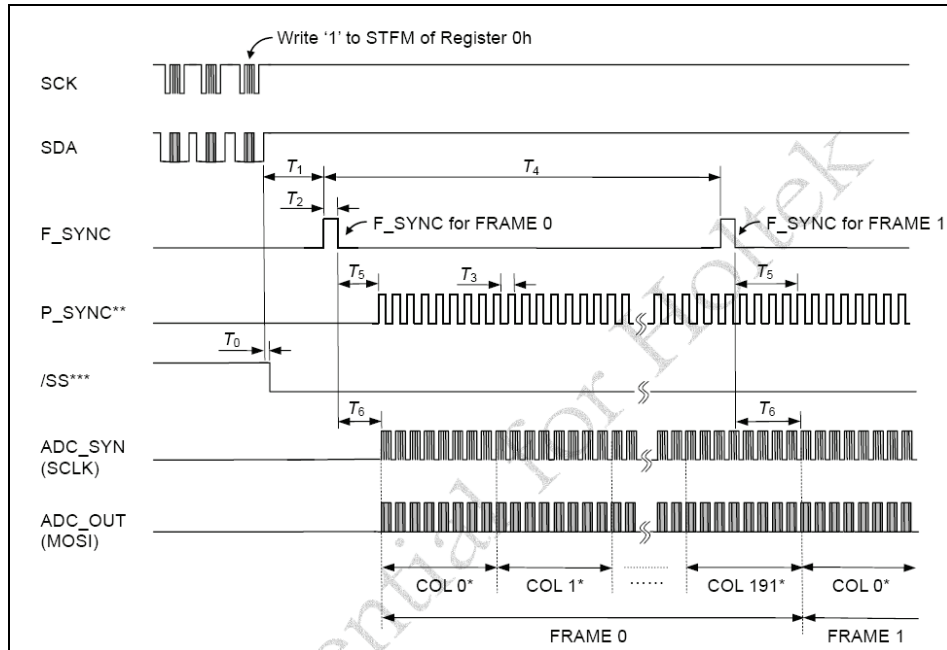
## 串列DMA介面指紋辨識模組應用

在串列 DMA 介面指紋辨識模組應用，使用祥群科技 LTT-SUPER S500 模組做為應用範例。

### 應用電路



應用時序圖



## 範例程式

```

//=====
// File name: main.asm
// [option]
// PB0/SCK      =      SCLK
// EP1          =      Enable
// EP2          =      Enable
// EP3          =      Enable
// SCLK         =      12Mhz
// SPI1_EN      =      Enable
// SPI1_Mode    =      mode=1
// WDT          =      Disable
// PA0/PSYNC,VSYNC = PSYNC or VSYNC
// PA5/PCLK     =      PCLK
// PA4/HSYNC    =      HSYNC
// 作者: Yen
// 說明: 該程式是主程式, 主要為做一些初始設定。
//       該程式使用 LTT-SS500A sensor
//=====
// 定義一些控制接腳
//=====
#define _CLK      [39h]      ; CLK Register
#define _SCLKEN   _CLK.0    ; PDMA 0: disable; 1: enable
#define _CLKAUTOB _CLK.1    ; SCLK halt 0: when FIFO3 is full or SCLKEN is 0
                          ; 1: when SCLKEN is 0

#define SDMAEN    MISC.3    ; SBDMA control bit 0: disable; 1: enable
#define SDMASEL   MISC.4    ; Serial DMA interface selection bit
                          ; 0: Serial interface 1 (pin-shared with port E)
                          ; 1: Serial interface 2 (pin-shared with port C)

#define SCLK_OUT  PBC.0
#define SW_RST_CTL PBC.5

#define CHIP_EN   PE.3
#define CHIP_EN_CTL PEC.3

#define I2C_SD_CTL PBC.7
#define I2C_SC_CTL PBC.6
#define I2C_SD     PB.7
#define I2C_SC     PB.6

//=====
// 設定USB Status
//=====
Set on_usb_clk      ; enable usb clock
Set set_v330        ; Turn on V330
clr ucc.6           ; 1=6M , 0=12M
clr USB_EP2_IO      ; Set EP2 OUTPUT FIFO
set USB_EP3_IO      ; Set EP3 INPUT FIFO

set SDMAEN

//=====
// 初始化控制信號
//=====

clr SCLK_OUT

clr SW_RST_CTL
    
```

```

clr I2C_SD_CTL
clr I2C_SC_CTL
SET CS_VOUT_CTL

clr SW_RST
clr I2C_SD
clr I2C_SC
call wait_20us
set SW_RST

SET SBCR.@M0          ;M0
SET SBCR.@M1          ;M1;slave
SET SBCR.@MLS         ;set, msb
clr SBCR.@SBEN

;=====
; // 設定MCU 資料抓取方式
;=====

CLR _CLKAUTOB
CLR SDMAEN
CLR SDMASEL
set _SCLKEN

;=====
; // 設定LTT-SS500
;=====
Call LTT-SS500_Init

;=====
; // interrupt enable
;=====
Set intc0.0           ; emi
Set intc1.0           ; usb

MainLoop:
sz suspend_line      ; USC.0=1, USB enter suspend mode
jmp Tosuspend        ; 處理 suspend
jmp MainLoop

;=====

;=====
; // File name: usb_int.asm
; // 說明: 本asm 檔為處理USB interrupt 資料
;=====
; // 處理 endpoint 1 資料
;=====

AccessFIFO1:
Sz nCmdIndex         ; Reset to 0
Jmp ExitExt1         ;

CALL FIFO1_Rd_Check   ; FIFO2 ready ?
Clr MISC.@USB_Req     ;
SZ Z                 ;
jmp ExitExt1         ;
out_fifo1_data:
CALL FIFO1_Rd_Check   ; FIFO2 ready ?
CALL ReadFIFO1        ;
    
```

```

//=====
// Enable Serial DMA
//=====
set SDMAEN          ; start SDMA
set SBCR.@SBEN     ; start SPI
ExitExt1:
Clr USR.@int1      ;
jmp Exit_of_Ext_start ;

//=====
// 處理 endpoint 2 資料
//=====
AccessFIFO2:
ExitExt2:
clr USR.@int2      ;
jmp Exit_of_Ext_start ;
//=====
// 處理 endpoint 3 資料
//=====
AccessFIFO3:
ExitExt3:
Clr USR.@int3      ;
Jmp Exit_of_Ext_start ;
    
```

### 程式說明

本範例主要是示範串列 DMA 的使用；使用 LTT-SS500，配合 HT82A525R USB MCU，完成取像並作指紋辨識的一個應用。

MCU 負責控制 Sensor，並利用第 1 組 SPI (PE0~PE4) 從 LTT-SS500 接收影像資料，透過 DMA 的方式，由 USB ENDPOINT 3 以 BULK 傳輸方式送到 PC 端，PC 端需自行開發 Vendor Class Driver，採用 BULK 傳輸方式接收資料。

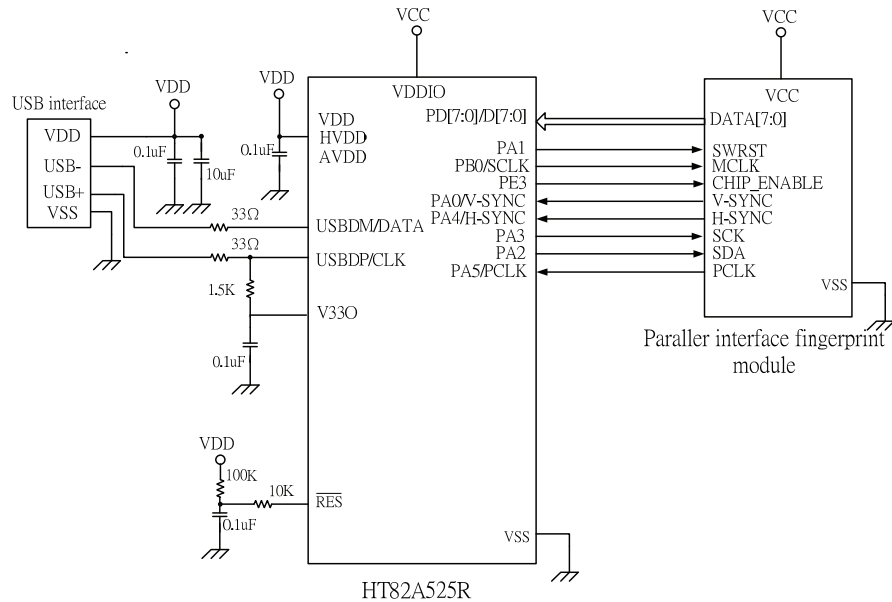
程式一開始先設定相關 Register，含 SPI 控制，因為 LTT-SS500 的 SPI 是固定為 Master 端，所以必須將 SPI 設定成 Slave，之後再初始化 Sensor，以 I<sup>2</sup>C 方式設定 Sensor，等待資料開始傳送。

在取影像前，本程式透過 PC 端對 ENDPOINT 1 下指令，先將 SDMA 致能及將 SCS 致能，打開串列 DMA 功能，開始接收傳送資料，此時 ENDPOINT 3 為 SDMA 模式，所以不用再手動處理資料，資料會以 DMA 的方式傳送給 PC 端。

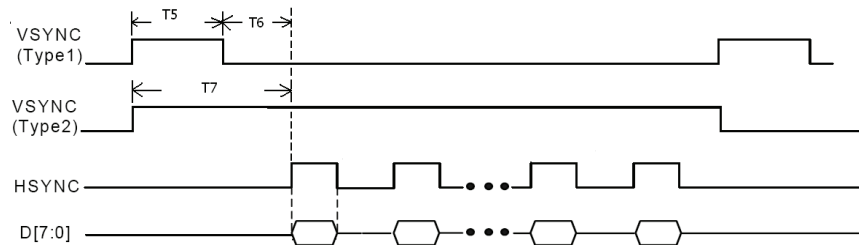
## 並列DMA介面指紋辨識模組應用電路

在並列 DMA 介面指紋辨識模組應用，使用 Hynix YACBAA0SDDAS 模組做為應用範例。

### 應用電路



### 應用時序圖



### 範例程式

```

//=====
// File name: main.asm
// PB0/SCK      =   SCLK
// EP1          =   Enable
// EP2          =   Enable
// EP3          =   Enable
// SCLK         =   12Mhz
// SPI1_EN     =   Enable
// SPI1_Mode   =   mode=1
// WDT         =   Disable
// PA0/PSYNC, VSYNC = PSYNC or VSYNC
// PA5/PCLK    =   PCLK
// PCLK        =   Falling Edge
// PA4/HSYNC   =   HSYNC

// 作者: Yen
// 說明: 該程式是主程式, 主要為做一些初始設定。
    
```

```

// 該程式使用 Hynix 702 cmos sensor
//=====
// 定義一些控制接腳
//=====
#define _CLK [39h] ; CLK Register
#define _SCLKEN_CLK.0 ; PDMA 0: disable; 1: enable
#define _CLKAUTOB_CLK.1 ; SCLK halt 0: when FIFO3 is full or SCLKEN is 0
; 1: when SCLKEN is 0
#define _PDMA_CLK.2 ; PDMA 0: disable; 1: enable
#define _PDATA_SEL_CLK.3; PDATA_SEL 0: odd byte 1: got all
#define _PDMA_MOD_CLK.4 ; PDMA_MOD 0: finger-print 1: normal pdma
#define _FIG_PIX_CLK.5 ; FIG_PIX 0: QVGA(320x240) 1: CIF(352x288)

#define SCLK_OUT PBC.0
#define I2C_SD_CTL PAC.2
#define I2C_SC_CTL PAC.3
#define I2C_SD PA.2
#define I2C_SC PA.3

#define SW_RST PA.1
#define CHIP_EN PE.3
#define SW_RST_CTL PAC.1
#define CHIP_EN_CTL PEC.3
//=====
// 設定 USB Status
//=====
set on_usb_clk ; enable usb clock
set set_v330 ; Turn on V330
clr ucc.6 ; 1=6M , 0=12M
clr USB_EP1_IO ; Set EP1 OUTPUT FIFO
set USB_EP2_IO ; Set EP2 INPUT FIFO
set USB_EP3_IO ; Set EP3 INPUT FIFO

//=====
// 初始化控制信號
//=====
clr SCLK_OUT
cClr CHIP_EN_CTL
clr SW_RST_CTL
clr I2C_SD_CTL
clr I2C_SC_CTL
clr I2C_SD
set I2C_SC
clr CHIP_EN_CTL
clr SW_RST_CTL

//=====
// chip enable
//=====
set CHIP_EN
call Delay4uSec
//=====
// enable MCLK
//=====
set _SCLKEN ; SCLKEN 0: SCLK disable 1: SCLK out enable
set _CLKAUTOB ; CLKAUTOB 0: stop by FIFO FULL 1: stop by SCLKEN
call Delay300mSec
    
```

```

//=====
// RESET Signal is High
//=====
set SW_RST
call Delay4uSec
//=====
// 設定MCU 資料抓取方式
//=====
clr _PDATA_SEL      ; 只取奇數筆資料
clr _PDMA_MOD       ; 並列DMA:指紋辨識模式
clr _FIG_PIX        ; QVGA (320x240) 模式

//=====
// 設定cmos sensor 值,影像大小,格式,時序等
//=====
call Sensor_Init

//=====
// interrupt enable
//=====
set intc0.0         ; emi
set intc1.0         ; usb

MainLoop:
sz suspend_line    ; USC.0=1, USB enter suspend mode
jmp Tosuspend      ; 處理suspend
jmp MainLoop

//=====

//=====
// File name: usb_int.asm
// 說明: 本asm 檔為處理USB interrupt 資料
//=====

//=====
// 處理 endpoint 1 資料
//=====

AccessFIFO1:
sz nCmdIndex       ; Reset to 0
jmp ExitExt1       ;

CALL FIFO1_Rd_Check ; FIFO2 ready ?
Clr MISC.@USB_Req  ;
SZ Z               ;
jmp ExitExt1       ;

out_fifo1_data:
CALL FIFO1_Rd_Check ; FIFO2 ready ?
CALL ReadFIFO1     ;

//=====
// Enable Parallel DMA
//=====
Set _PDMA

ExitExt1:
Clr USR.@int1      ;
jmp Exit_of_Ext_start ;
    
```

```

//=====
// 處理 endpoint 2 資料
//=====
AccessFIFO2:
ExitExt2:
    Clr  USR.@int2          ;
    Jmp  Exit_of_Ext_start ;
//=====
// 處理 endpoint 3 資料
//=====
AccessFIFO3:
ExitExt3:
    clr  SR.@int3          ;
    jmp  xit_of_Ext_start  ;
    
```

## 程式說明

本範例主要是示範並列 DMA 的使用；使用 Hynix CMOS Sensor (HI-702)，配合 HT82A525R USB MCU，完成取像並作指紋辨識的一個應用。

MCU 負責控制 Sensor，並利用 PD0~PD7 接收影像資料，透過 DMA 的方式，由 USB ENDPOINT 2 以 BULK 傳輸方式送到 PC 端，PC 端需自行開發 Vendor Class Driver，採用 BULK 傳輸方式接收資料。

程式一開始初始化 Sensor，由 I<sup>2</sup>C 方式控制 Sensor Register 設定 Sensor 的模式，如影像資料格式、大小及頻率等。因為指紋辨識不需使用到彩色的資料，Hi-702 資料格式有 RGB 及 YUV 兩種格式，一個圖元二個 byte，若只接收 Y 或三原色中的一種原色，也可以順利還原成單色影像，所以為了 MCU 可將 PDATA\_SEL 設定為 1，允許只接收奇數筆的資料，加快速度。

另外，為配合市面上常見的影像格式 QVGA 及 CIF，可藉由 FIG\_PIX 的設定，設定 MCU 目前所要處理影像為 QVGA 或者 CIF 的格式。若將 PDMA\_MOD 設定為 0，則表示為指紋辨識模式，若設定為 1，則表示為正常模式。在指紋模式下，MCU 內部會根據 VSYNC 及 HSYNC 的訊號並控制 MCLK，自動判斷每張影像的開始，如此一來，可確保 MCU 所抓取的資料，每次都從影像的第一筆資料開始抓取，避免發生影像偏差的情形，若為正常模式，則 MCU 會一直收取資料，不會自動判斷起始點，適合一般性的資料。

在取影像前，本程式透過 PC 端對 ENDPOINT 1 下指令，先將 PDMA 致能，打開並列 DMA 功能，開始接收傳送資料，此時 ENDPOINT 2 為 DMA 模式，所以不用再手動處理資料，資料會以 DMA 的方式傳送給 PC 端。

HT82A525R 可藉由 PDATA\_SEL、PDMA\_MOD、FIG\_PIX 設定，適用於大部份的並列 DMA 應用。

## 版本記錄

**版本：V1.10**

修改人員：謝琳

修改日期：2012 年 1 月 9 日

修改內容：

- 應用電路圖中 USB $\bar{D}$ -引腳與 USBDM/DATA 引腳間、USB $\bar{D}$ +引腳與 USBDP/CLK 引腳間多加了一個 33 $\Omega$ 電阻， $\overline{\text{RES}}$  引腳與 0.1 $\mu\text{F}$  電容間多加了一個 10k $\Omega$ 電阻。
- 電路圖中 USB $\bar{D}$ - / USB $\bar{D}$ + 改為 USBDM / USBDP，RES 改為  $\overline{\text{RES}}$ 。